1Л

УДК 004.421.6.

М.С. Львов
Херсонский государственный университет, Украина
lvov@ksu.ks.ua

Интеллектуальные свойства систем компьютерной математики учебного назначения и методы их реализации

В работе сформулированы основные задачи систем компьютерной математики учебного назначения, реализация которых обеспечивает интеллектуальные свойства этих систем. Описаны методы решения задач верификации хода решения учебной математической задачи в различных педагогических ситуациях, использующие технологии символьных преобразований и алгоритмы компьютерной алгебры.

Введение

Точные и естественные дисциплины занимают особое место среди учебных дисциплин. Они формируют фундаментальные научные знания, основанные на точных математических моделях явлений и процессов в природе и обществе. Процесс обучения этим дисциплинам включает не только лекции, но и активные формы обучения: практические занятия, лабораторные работы и т.п.

Практическая математическая деятельность ученика – основная форма учебной деятельности. Она заключается в решении учебных математических задач.

Учебная цель практической работы – построение хода решения учебной задачи, а не только ответа. Поэтому системы учебного назначения по математике, далее называемые *системами компьютерной математики учебного назначения* (СКМУН), должны поддерживать именно процесс решения учебной математической задачи [1-3].

Контроль процедурных знаний – одна из основных задач СКМУН, обладающих нтеллектуальными чертами. Технологии контроля процедурных знаний исследованы и разработаны еще недостаточно. *Предмет настоящей работы – формулировка задач интеллектуальной поддержки практической математической деятельности и методов их решения в СКМУН.*

Под СКМУН мы понимаем программные системы учебного назначения по дисциплинам, которые используют математические модели и методы соответствующих предметных областей, основанные на технологиях символьных преобразований и методах компьютерной алгебры.

1 Функциональные требования к интеллектуальным свойствам СКМУН

Информационная поддержка процесса решения учебной задачи возможна при условии, что ее решение происходит в специализированном программном модуле – деятельностной среде (ДС).

Один из наиболее важных аспектов поддержки практической математической деятельности ученика – *проверка правильности выполнения его действий на разных этапах решения задачи* – начиная от этапа построения математической модели и заканчивая этапом проверки правильности хода решения или ответа. Второй, не менее важный аспект поддержки, – *автоматизация рутинных действий*, *связанных с вычислениями*. Третий аспект – *предоставление ученику удобной системы подсказок на разных этапах решения задачи в виде генерации математической модели задачи, хода или шага ее решения, ответа.*

Практическая деятельность преподавателя также должна поддерживаться. Первый аспект такой поддержки – *проверка правильности хода решения задачи*. Система должна проверять правильность хода решения задачи, решенной учеником (режим проверки контрольной работы). Второй аспект – *автоматизация тестирования знаний учеников.*

Для реализации этих функций предназначены разные типы деятельностных сред. В частности, это *среда тестирования*, *среда решения задач.*

Основные системные задачи среды тестирования – задача генерации тестовых заданий и задача проверки правильности ответов. Их решение излагается в [4].

Основные системные задачи среды решения (СРЗ) – это функции поддержки процесса решения учебной математической задачи. Разные аспекты этой поддержки определяют различные режимы СРЗ. Перечислим эти задачи:

1. Задача верификации модели учебной задачи (УЗ). Широкий класс УЗ требует от пользователя самостоятельного построения модели УЗ. Это, например, так называемые текстовые задачи курса школьной алгебры, решаемые с помощью уравнений или систем уравнений. Задача верификации состоит в проверке правильности модели, составленной пользователем, т.е. ее эквивалентности правильной модели.

2. Задача верификации шага решения учебной задачи. Решая УЗ по шагам, пользователь может допустить ошибку на каждом шаге решения. Задача верификации шага решения состоит в проверке правильности преобразования модели, выполненной пользователем на данном шаге.

3. Задача верификации хода решения учебной задачи. Решив УЗ при выполнении контрольной работы, пользователь должен представить ход решения УЗ для проверки. Проверку осуществляет преподаватель на своем рабочем месте в режиме *offline*. Если ошибка в ходе решения найдена, это не должно привести к окончанию проверки. Задача заключается в обнаружении всех ошибок в ходе решения и их фиксации.

4. Задача генерации шага решения УЗ. В некоторых педагогических ситуациях пользователю, решающему УЗ, система должна предоставить подсказку в виде очередного шага решения УЗ.

5. Задача генерации хода решения УЗ. В некоторых педагогических ситуациях система должна предоставить пользователю методически правильный ход решения УЗ. Алгоритмы решения этой задачи реализованы во многих СКМ.

6. Задача автоматической поддержки хода решения УЗ. Решение этой задачи реализовано во многих коммерческих СКМУН.

Ниже мы рассмотрим методы решения системных задач верификации для алгебраических задач.

2 Модель дидактического содержания СКМУН

Основная структурная единица учебного материала математической дисциплины – учебный модуль. Приведем формальное определение этого понятия:

Сигнатура учебного модуля. Математические теории, которые излагаются в учебном модуле, используют, как правило, новые математические символы. Например, учебный

модуль *Тригонометрия* вводит символы тригонометрических и обратных тригонометрических функций, символ константы π. Предметом изучения являются интерпретации символов сигнатуры. Проблемам реализации интерпретаторов сигнатур посвящены работы [5], [6].

Математические модели учебного модуля. К математическим моделям УМ принадлежат формальные определения математических объектов, являющихся предметом изучения. В модуле *Тригонометрия* это, например, формальные определения тригонометрического выражения, тождества, уравнения, неравенства.

Типы учебных задач учебного модуля. Основной предмет изучения учебного модуля математической дисциплины – учебные задачи, перечень типов которых определен в программе дисциплины. Наше определение учебной задачи включает: модель $M(x_1,...,x_n)$, условие задачи $\varphi(x_1,...,x_n)$ и вопрос $Q(x_{j_1},...,x_{j_m})$ к ней. Формальное определение стандартной задачи можно интерпретировать как

$$\textit{Дано } M(x_1,...,x_n), \textit{ причем } \varphi(x_1,...,x_n). \textit{ Найти } Q(x_{j_1},...,x_{j_m}).$$

Например, задача «Построить касательную L к графику F функции $y=\frac{x+1}{x}$ в точке A с абсциссой $x_A=1$» представлена в виде модели $P=<M,\varphi,Q>$, где

$$M=F(y=f(x)\ \&\ A(x_A,y_A)\ \&\ L(y-y_A=f'(x_A)(x-x_A)),$$
$$\varphi=(f(x)=\frac{x+1}{x})\ \&\ (x_A=1), \tag{1}$$
$$Q=L.$$

Модель задачи использует модели графика функции $y=f(x)$, точки $A(x_A,y_A)$ и касательной к графику функции $L(y-y_A=f'(x_A)(x-x_A))$. Эти модели объединены соотношениями $(f(x)=\frac{x+1}{x})\ \&\ (x_A=1)$, определенными в условии задачи.

Элементарные преобразования моделей. Процесс решения учебной задачи определяется как последовательность шагов, на каждом из которых осуществляется одно из элементарных преобразований $M\ \&\ \varphi$. В каждом учебном модуле определены специфические преобразования. Например, специфическими для учебного модуля *Дифференциальное исчисление* являются правила дифференцирования и таблица производных элементарных функций.

Определение учебного модуля. Учебный модуль SD описывается классом учебных задач $Task_{SD}$, который и определяет содержание SD. Учебные задачи определены в терминах математических моделей MM_{SD} и отношений зависимости. MM_{SD} определены в терминах сигнатур Σ_{SD} учебного модуля и элементарных преобразований ET_{SD}. Таким образом, собственно SD определяется как четверка

$$SD=<\Sigma, MM, ET, Task>.$$

Анализ СЛС таких математических дисциплин, как школьная алгебра, математический анализ, линейная алгебра, теория обычных дифференциальных уравнений, некоторых других дисциплин показал, что все они удовлетворяют этой схеме. Соответствующая СКМНП может строиться как единая система, основанная на понятии *алгебраического объекта* (АО) и *эквационального вывода*.

Модель учебной алгебраической задачи. Под учебной алгебраической задачей (УАЗ) мы понимаем задачу, формулируемую в терминах алгебраических объектов, являющуюся предметом изучения и поддерживаемую СКМУН.

Определение 2.1. Сигнатурой Σ предметной области называется пара $<\Sigma_O, \Sigma_p>$, где Σ_o – сигнатура операций, Σ_p – сигнатура предикатов.

Определение 2.2. Примитивным алгебраическим объектом называется элемент носителя соответствующей многосортной алгебраической системы или переменная.

Определение 2.3. Атомарным алгебраическим объектом называется терм, составленный из примитивных АО в сигнатуре Σ_o.

Определение 2.4. Структурированным алгебраическим объектом (САО) в сигнатуре Σ называется бескванторная формула прикладной логики предикатов $F(x_1,...,x_n)$ с равенством. Множество переменых $\{x_1,...,x_n\}$ называется *координатным множеством* или *координатным пространством* САО.

Сигнатура Σ_p атомарных предикатов формулы $F(x_1,...,x_n)$, кроме предикатов равенства и отрицания равенства $(\neq)$, может содержать и другие атомарные предикаты, определение и интерпретация которых осуществляется в каждой предметной области.

В частности, для «школьных» САО это предикаты строгого и нестрогого порядка. Логические связки – конъюнкция и дизъюнкция.

Алгебраический тип УАЗ определяется типами элементов ее структуры. Например, тип УАЗ «Решить тригонометрическое уравнение» определен структурой алгебраических типов (рис. 1).

Эта структура уточняет условие задачи в терминах алгебр, используемых при решении УАЗ. В рассматриваемом примере структура УАЗ определяет рациональное тригонометрическое уравнение одного неизвестного, коэффициенты которого принадлежат полю $TCoef$, аргументы тригонометрических функций – алгебре $TArg$, а решение ищется в виде элемента алгебры $TSol$.

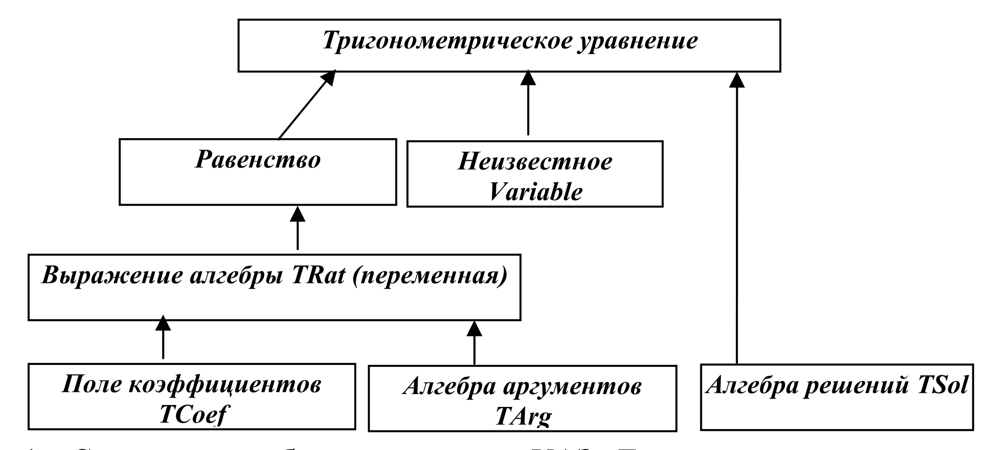


Рисунок 1 – Структура алгебраических типов УАЗ «Тригонометрическое уравнение»

Определение 2.5. Учебной алгебраической задачей называется структурированный алгебраический объект с определенным для него алгебраическим типом.

В качестве модели алгебраических вычислений, реализующих сигнатуры предметных областей, используется понятие упорядоченно-сортной алгебраической системы (МАС) [7], [8]. Уточнение этого понятия, используемое в спецификациях СКМУН, описано в [5], [6].

3 Задачи верификации и методы их решения

I. Задача верификации модели учебной задачи. Рассмотрим эту задачу для класса учебных алгебраических задач. Пусть $P =< \mathrm{M}(x_1,...,x_n), \varphi(x_{i_1},...,x_{i_k}), Q(x_{j_1},...,x_{j_m}) >$ – формулировка УАЗ. Как мы отмечали, для алгебраических задач ее модель M и условие φ объединены в формуле $F(\mathrm{x}_1,...,\mathrm{x}_\mathrm{n}) = \mathrm{M}(\mathrm{x}_1,...,\mathrm{x}_\mathrm{n})\ \&\ \varphi(x_{i_1},...,x_{i_k})$.

Вопрос $Q(x_{j_1},...,x_{j_m})$ имеет вид $(x_{j_1},...,x_{j_m}) \in ?$ (найти $(x_{j_1},...,x_{j_m})$). Это означает, что, вообще говоря, не все переменные модели задачи являются неизвестными. Решение задачи – это некоторое подмножество $A = \{(x_{j_1}^{(i)},...,x_{j_m}^{(i)}) \mid F(\mathrm{x}_1,...,\mathrm{x}_\mathrm{n}), i = 0,1,2,...\}$.

Пусть $Solve(F)$ – алгоритм решения задачи типа F, генерирующий A в определенной канонической форме выражений из алгебры $Sol(x_{j_1},...,x_{j_m})$: $Can_{Sol}(A) = A$. Пусть, далее, $F'(y_1,...,y_m)$ – модель, построенная пользователем и $Solve(F')$ – ее решение. Тогда верификация состоит в проверке

$$(x_{j_1},...,x_{j_m}) \subseteq (y_1,...,y_m)\ \&\ Solve(F'(y_1,...,y_m))|_{(x_{j_1},...,x_{j_m})} = A\,. \qquad (2)$$

Другими словами, ограничение решения модели УАЗ, построенной пользователем, на подмножество ее неизвестных, должно быть равно канонической форме ответа УАЗ. Алгоритм верификации модели использует:

1) список неизвестных (переменных, значения которых ищутся). Этот список задает составитель задачи в текстовой формулировке условия задачи;

2) эталонную модель задачи $F(\mathrm{x}_1,...,\mathrm{x}_\mathrm{n})$. Модель задает составитель задачи в формулировке условия задачи;

3) решение задачи A, предоставленное в канонической форме. Это решение составитель получает, решая задачу в среде вычислений (функция Solve).

Исходные данные 1 и 2 алгоритма верификации модели вводятся в систему в специализированном редакторе *Задачника* системы. К редактору имеют доступ авторы задач (методист и учителя).

Свое решение ученик формулирует в специальном окне *Модель задачи* СРЗ, которое открывается командой *Начать решение*. Команда *Проверить* вызывает функцию проверки правильности модели (1). Приведем пример:

Составить модель задачи:

Текстовое условие задачи	Эталонная модель
Поезд на середине пути между станциями А и В задержался на 15 мин. Чтобы прибыть на станцию В вовремя, машинист увеличил скорость движения на 10 км/час. Найти скорость поезда V, если расстояние между станциями составляет 210 км.	$\frac{105}{V} + \frac{15}{60} + \frac{105}{V+10} = \frac{210}{V}$

Формулировка задачи:

$$MM(S,V,v,t) = (\frac{S}{2V} + \frac{t}{60} + \frac{S}{2(V+v)} = \frac{S}{V})\ \&\ (V > 0)\,,$$

$$\varphi = (S = 105)\ \&\ (t = 15)\ \&\ (v = 10)\,,$$

$$Q(S) = (S = ?)\,.$$

Модель задачи – уравнение $\frac{S}{2V}+\frac{t}{60}+\frac{S}{2(V+v)}=\frac{S}{V}$, из которого нужно найти положительные значения V. Условие задачи – численные значения параметров S, v, t. Имя неизвестной величины V обязательно должно быть указано в условии.

Эталонная модель, построенная системой по условию задачи:

$$F(V)=\frac{105}{V}+\frac{15}{60}+\frac{105}{V+10}=\frac{210}{V}.$$

Модель, построенная пользователем:

$$F^{'}(V)=(t_1=\frac{105}{V})\,\&\,(t_2=\frac{105}{V+10})\,\&\,(t_1+\frac{1}{4}+\mathrm{t}_2=2\mathrm{t}_1)\ .$$

Решение $F(V)$: $((V=60)\vee(V=-70))\,\&\,(V>0)\sim(V=60)$.

Решение $F^{'}(t_1,t_2,V)$: $((V=60)\vee(V=-70))\,\&\,(V>0)\,\&\,(t_1=\frac{7}{4})\,\&\,(t_2=\frac{3}{2})$.

Ограничение $F^{'}(V)$ решения: $((V=60)\vee(V=-70))\,\&\,(V>0)\sim(V=60)$.

Результат проверки: $(V=60)\sim(V=60)\sim True$.

Наличие в условии задачи скрытой от ученика модели задачи дает возможность, во-первых, автоматизировать процесс проверки правильности модели и ответа, во-вторых, автоматизировать процесс тестирования деятельностных сред и программного модуля *Задачник,* в-третьих – реализовать функцию подсказки ученику, отображающую на экран правильную модель учебной задачи.

Определение 2.4 позволяет сформулировать алгоритм $Solve(F)$ для базового типа формул $F(x_1,...,x_n)$. Именно, пусть

$$\Sigma_O=<()+(),()-(),()*(),()/(),()\hat{}Rat,\sqrt[Nat]{()},|()|>,\ \Sigma_p=<()=(),()\neq(),()<(),()>(),()\in Nat>.$$

Тогда F можно представить в виде дизъюнктивной нормальной формы $F = S_1\,\&\,U_1\vee...\vee S_k\,\&\,U_k$, где $F_j(X)$ – системы целых алгебраических уравнений, а $U_j(X)$ – системы целых алгебраических неравенств и, возможно, дополнительных условий вида $g(X)\in Nat$. Алгоритм $Solve(F)$ заключается в решении систем уравнений $F_j(X)$ методом построения базисов Гребнера и последующих проверок $U_j(X)$ на полученных решениях.

II. *Задача верификации шага решения учебной задачи.* Пусть $M(A)$ – модель УЗ, A – алгебраический объект, выделенный пользователем в $M(A)$, и $A^{'}$ – преобразование A, выполненное пользователем. Тогда необходимо проверить $A\overset{M}{=}A^{'}$. Алгоритм проверки $A\overset{M}{=}A^{'}$ зависит от типа выделенного подвыражения в структуре $M(A)$:

1. A – число. Тогда $A^{'}$ – числовое выражение, такое, что $Val(A^{'})=A$.

2. A – переменная. Тогда $A^{'}$ – алгебраическое выражение, такое, что $Val(A^{'})=A$.

3. A – терм в сигнатуре Σ_O. Тогда $A^{'}$ – терм, такой, что $Val(A^{'})=A$.

4. A – атомарное логическое выражение. Если это уравнение или неравенство одной переменной, то $A^{'}$ – также логическое выражение, такое, что $Solve(A^{'})=Solve(A)$.

5. A – структурированное логическое выражение нескольких переменных. Тогда A' – также логическое выражение, такое, что $Solve(A') = Solve(A)$.

Итак, для реализации алгоритма задачи необходимо реализовать функцию $Val(A)$, удовлетворяющую условиям (1 – 3). Для широкого класса выражений это можно сделать методами построения канонических форм, изложенными в [9].

Алгоритм функции $Solve(F)$ изложен в предыдущем пункте.

III. Задача верификации хода решения учебной задачи. В обозначениях предыдущей задачи алгебраическая ошибка определяется соотношением $A \overset{M}{\neq} A'$. Поскольку наличие ошибки не должно прерывать проверку хода решения, ошибки следует классифицировать. Наша точка зрения заключается в следующем:

1. Существуют фатальные и текущие ошибки. Фатальные ошибки – это ошибки, нарушающие алгебраический тип или координатное множество задачи. Например, если на некотором шаге пользователь преобразовал уравнение к терму ($f(x) = 0 \sim g(x)$), он допустил фатальную ошибку и дальнейшая проверка не имеет смысла.

2. Пусть $Var(A)$ обозначает координатное множество переменных выражения A. Текущие ошибки классифицируются по типам выделенных подвыражений:

1) A – число. Тогда A' – числовое выражение, такое, что $Val(A') = A$. Если $Val(A') \neq A$, допущена арифметическая ошибка;

2) A – переменная. Тогда A' – алгебраическое выражение, такое, что $Val(A') = A$. Если $Val(A') \neq A$, но $Var(A') \subseteq Var(M)$, допущена алгебраическая ошибка;

3) A – терм в сигнатуре Σ_O. Тогда A' – терм, такой, что $Val(A') = Val(A)$. Если $Val(A') \neq Val(A)$, но $Var(A') \subseteq Var(M)$, допущена алгебраическая ошибка;

4) A – атомарное логическое выражение. Если это уравнение или неравенство одной переменной, то A' – также логическое выражение, такое, что $Solve(A') = Solve(A)$. Если $Solve(A') \neq A$, но $Var(A') \subseteq Var(M)$, допущена ошибка метода решения;

5) A – структурированное логическое выражение нескольких переменных. Тогда A' – также логическое выражение, такое, что $Solve(A') = Solve(A)$. Если $Solve(A') \neq$ $\neq Solve(A)$, но $Var(A') \subseteq Var(M)$, допущена логическая ошибка.

Выводы

Основная цель систем компьютерной математики учебного назначения – поддержка практической учебной деятельности, которая заключается в решении учебных математических задач. Для достижения этой цели СКМУН должны обладать рядом специфических интеллектуальных свойств, которые формулируются в виде системных задач. К интеллектуальным свойствам относится, в частности, способность системы осуществлять проверку правильности хода решения учебной задачи, решаемой пользователем в системе. Реализация системных задач верификации позволяет сделать эффективной самостоятельную работу ученика в СКМУН и в значительной степени освободить учителя от рутинной работы по контролю практической работы учеников.

Литература

1. Львов М.С. Методи проектування систем комп’ютерної підтримки математичної освіти / М.С. Львов, О.В. Співаковський // Математичні моделі і сучасні інформаційні технології : зб. наук. праць НАН України. – Київ, 1998 . – С. 101-111.

2. Львов М. Основные принципы построения педагогических программных средств поддержки практических занятий / М. Львов // Управляющие системы и машины. – 2006. – № 6. – С. 70-75.
3. Львов М.С. Шкільна система комп'ютерної алгебри ТерМ 7-9. Принципи побудови та особливості використання / М.С. Львов // Науковий часопис НПУ ім. Драгоманова: зб. наук. праць. – К. : НПУ ім. Драгоманова, 2005. – № 3(10). – С. 160-168. – (Серія № 2: «Комп'ютерно-орієнтовані системи навчання»).
4. Львов М.С. Математические тесты в системах компьютерной математики учебного назначения. / М.С. Львов // Управляющие системы и машины. – 2011 (в печати).
5. Львов М.С. Синтез інтерпретаторів алгебраїчних операцій в розширеннях багатосортних алгебр / М.С. Львов // Вісник Харківського національного университету. – 2009. – № 847. – С. 221-238. – (Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»).
6. Львов М.С. Об одном подходе к реализации алгебраических вычислений: вычисления в алгебре высказываний / М.С. Львов // Вестник Харк. нац. ун-та. – 2009. – № 863. – С. 157-168. – (Серия «Математическое моделирование. Информационные технологии. Автоматизированные системы управления»).
7. Goguen J. Ordered-Sorted Algebra I: Partial and Overloaded Operations. Errors and Inheritance / J. Goguen, J. Meseguer // Theoretical Computer Science. – Oxford : Elsevier. – 1992. – Vol. 105, № 2. – P. 217-273.
8. Goguen J.A. An initial algebra approach to the specification, correctness and implementation of abstract data types / J.A.Goguen, J.W. Thatcher, E. Wagner // Current Trends in Programming Methodology. – Englewood Cliffs, NJ : Prentice Hall, 1978. – P. 80-149.
9. Аржанцев И.В. Базисы Грёбнера и системы алгебраических уравнений / Аржанцев И.В. – Изд-во МЦНМО, 2003. – 68 с.
10. Львов М.С. Проектирование логического вывода как пошагового решения задач в математических системах учебного назначения / М.С. Львов // Управляющие системы и машины. – 2008. – № 1. – С. 25-32.
11. Львов М.С. Математичні моделі та методи підтримки ходу розв'язання навчальних задач з аналітичної геометрії / М.С. Львов // Искусственный интеллект. – 2010. – № 1. – С. 86-92.
12. Львов М.С. Математические модели предметных областей в системах компьютерной математики учебного назначения / М.С. Львов // Матеріали конференції «ІКТ в освіті, дослідженнях та індустриальних додатках: інтеграція, гармонізація та трансфер знань» (4-8 травня 2011 р.). – Херсон : вид.ХДУ, 2011. – С. 95-96.
13. Бухбергер Б. Базисы Гребнера. Алгоритмический метод в теории полиномиальных идеалов / Б. Бухбергер // Компьютерная алгебра: символьные и алгебраические вычисления / [под ред. Б. Бухбергера, Дж. Коллинза, Р. Лооса.]. – М. : Мир, 1986. – С. 331-383.

Literatura

1. L'vov M.C. Matematychni modeli i suchasni informacijni tehnologii:zb. nauk. prac' NAN Ukraini. Kiev. 1998 . P. 101-111.
2. L'vov M. Upravljajushhie sistemy i mashiny. 2006. №6. P. 70-75.
3. L'vov M.S. Naukovij chasopis NPU im.Dragomanova: zb.nauk. prac'. Kiev : NPU im. Dragomanova. 2005. № 3(10). P. 160-168.
4. L'vov M.C. Upravljajushhie sistemy i mashyny. 2011(in printing).
5. L'vov M.S. Upravljajushhie sistemy i mashyny. 2008.№ 1. P. 25-32.
6. L'vov M.S. Iskustvennyj intellekt. 2010. №1. P. 86-92.
7. L'vov M.S. Materiali konferencii"IKT v osviti, doslidzhennjah ta industrial'nih dodatkah: integracija, garmonizacija ta transfer znan'" May 4-8, 2011. Herson: vid.HDU. 2011. P. 95-96.
8. Goguen J. Oxford: Elsevier. 1992.Vol.105 (Num 2). P. 217-273.
9. Goguen J.A. Prentice Hall. 1978. P. 80-149.
10. L'vov M.S. Visnik Har'kivs'kogo nacional'nogo universitetu. 2009.№ 847. P. 221-238.
11. L'vov M.C. Vestnik Hark. nac. un-ta. 2009. № 863. P. 157-168.
12. Buhberger B. Moscow: Mir. 1986. P. 331-383.
13. Arzhancev I.V. Izd-vo MCNMO. 2003. 68 p.

M.S. Lvov

Intellectual Properties of Computer Mathematics Systems of Educational Purpose and Methods of Their Realization

In the article the basic tasks of the systems of computer mathematics of educational purpose are defined, the realization of which provides intellectual properties of these systems. The methods of solving of tasks of verification of course of solving of educational mathematical task in different pedagogical situations, using technologies of symbol transformations and algorithms of computer algebra are described.

Статья поступила в редакцию 10.06.2011.